

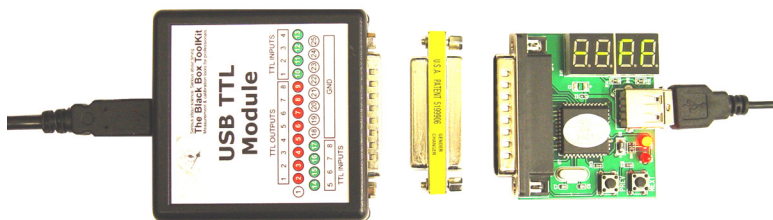


Serious about science: Serious about timing
The Black Box ToolKit
 Measurement & calibration tools for professionals

USB TTL Event Marking Module Diagnostic Board Manual

The USB TTL Module Diagnostic Board allows you to quickly and easily check event marks generated from your own software and scripts without external equipment connected.

The Diagnostic Board plugs into the 25-way socket of the TTL Module via a Gender Changer or using a 25-way Cable.



To check the correct outputs are being generated plug the USB lead into the TTL Module as normal. Next plug the USB power lead into the Diagnostic Board and into a second USB port on your PC. If you connect the diagnostic board directly to the TTL Module via the Gender Changer you may need to flip the Module over as shown.

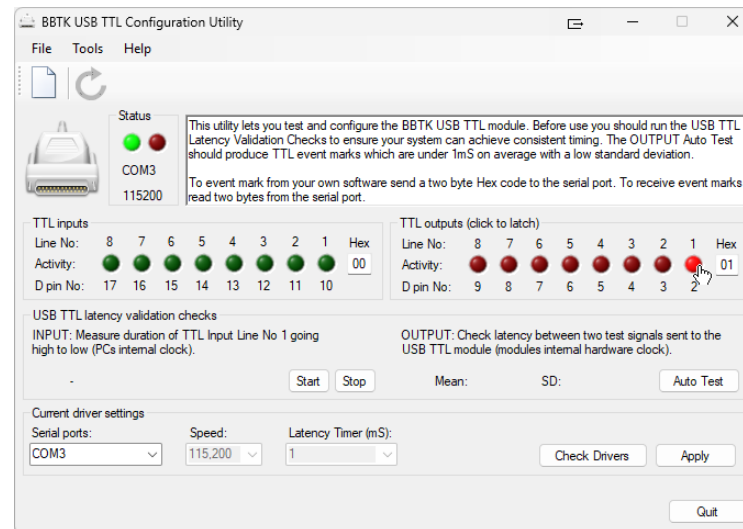


When you produce an Event Mark, or Trigger, you output two Hex Bytes from your own software. This then sends individual pins or lines, high (+5V) or low (0V), on the TTL Module. Each time you generate an Event Mark the Diagnostic Board looks at the physical TTL or +5V signals on each pin and converts them back into Hexadecimal. This enables you to check your scripts are outputting the correct Hex and that the correct pins are going high and low to event mark on your own equipment.

As you output from your own scripts to send given pins, or lines, high or

low, each pair of Hex bytes will scroll from right to left on the two LED readouts. You may need to slow your outputs down so you can see them.

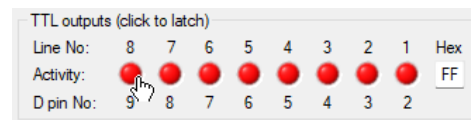
For example using the TTL Module Configuration Utility if all eight output pins are working (D pin No's 2~9 or Line No's 1 to 8) the Hex output should scroll from right to left as you click on each output LED in turn.



The sequence of Hex bytes that should scroll past on the diagnostic board LED's are as follows:

01 03 07 0F 1F 3F 7F FF

FF is the final state where all 8 pins, or lines, are high (+5V).



Two small push buttons labelled PREV and NEXT on the diagnostic board allow you to scroll through the output stream.

